International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-3, Issue-3, May 2015

Security Issue on "Secure Data Storage and Transaction Logs" In Big Data

Mr. Yannam Apparao, Mrs. Kadiyala Laxminarayanamma

***Abstract*- paper proposed the security issue on secure data storage and transaction logs in associated with data storage, management, and processing, though security not available in various things in big data unauthorized access may corrupt the data, this paper provides the security on transaction logs are transfer the information from once source to another source while transferring the data unauthorized access may tamper and forwards therefore users may not get exact data. Hence the paper proves security issue on secure data storage and transaction logs.**

***Keyword*- big data, transaction logs, data storage.**

I. INTRODUCTION

Big data is an evolving term that describes any voluminous amount of structured, semi-structured and unstructured data that has the potential to be mined for information. Although big data doesn't refer to any specific quantity, the term is often used when speaking about petabytes and exabytes of data. "Big Data" as the amount of data just beyond technology's capability to store, manage and process efficiently. These imitations are only discovered by a robust analysis of the data itself, explicit processing needs, and the capabilities of the tools (hardware, software, and methods) used to analyze it. As with any new problem, the conclusion of how to proceed may lead to a recommendation that new tools need to be forged to perform the new tasks. As little as 5 years ago, we were only thinking of tens to hundreds of gigabytes of storage for our personal computers. Today, we are thinking in tens to hundreds of terabytes. Thus, big data is a moving target another way, it is that amount of data that is just beyond our immediate grasp, e.g., we have to work hard to store it, access it, manage it, and process it.

The current growth rate in the amount of data collected is staggering. A major challenge for IT researchers and practitioners is that this growth rate is fast exceeding our ability to both: (1) designs appropriate systems to handle the data effectively and (2) and analyze it to extract relevant meaning for decision making. In this paper identify the Security issues on secure data storage and transaction logs in associated with data storage, management, and processing.

Manuscript received April 15, 2015

Mr. Yannam Apparao, Currently working as Associate Professor, Marri Laxman Reddy Institute of Technology and Management, Dundigal, Quthbullar(M),R. R. Distic, Telangana-500043,India.

Mrs. K.Laxminarayanamma, Currently working as Associate Professor, Institute of Aeronautical Engineering, Quthbullapur(M), R. R. Distic, Telangana, India.

II.RELATED WORK

Traditional security mechanisms, which are tailored to securing small-scale static (as opposed to streaming) data, are inadequate. For example, analytics for anomaly detection would generate too many outliers. Similarly, it is not clear how to retrofit provenance in existing cloud infrastructures. Streaming data demands ultra-fast response times from security and privacy solutions.In this paper, we highlight the top ten big data specific security and privacy challenges. We interviewed Cloud Security Alliance members and surveyed security practitioner-oriented trade journals to draft an initial list of high-priority security and privacy problems, studied published research, and arrived at the following top ten challenges:

- Secure data storage and transactions logs
- Real-time security/compliance monitoring
- Secure computations in distributed programming frameworks
- Security best practices for non-relational data stores
- End-point input validation/filtering
- Scalable and compassable privacy-preserving data mining and analytics
- Cryptographically enforced access control and secure communication
- Granular access control
- Granular audits
- Data provenance

III. SECURE DATA STORAGE AND TRANSACTIONS LOGS

Big data storage system with the attributes required will often be scale-out or clustered NAS. This is file access shared storage that can scale out to meet capacity or increased compute requirements and uses parallel file systems that are distributed across many storage nodes that can handle billions of files without the kind of performance degradation that happens with ordinary file systems as they grow.

Data and transaction logs are stored in multi-tiered storage media. Manually moving data between tiers gives the IT manager direct control over exactly what data is moved and when .However, as the size of data set has been, and continues to be, growing exponentially, scalability and availability have

necessitated auto-tiering for big data storage management. Auto-tiering solutions do not keep track of where the data is stored, which pose s new challenges to secure data storage. New mechanisms are imperative to thwart unauthorized access and maintain the 24/7 availability.here we provided the auto-tiering mechanism.

Tiered storage is the assignment of different categories of data to different types of storage media in order to reduce total storage cost. Categories may be based on levels of protection needed, performance requirements, frequency of use, and other considerations. Since assigning data to particular media may be an ongoing and complex activity,

A. Secure Data Storage in big data

Our storage securities solutions help prevent unauthorized modification or disclosure of data stored across your enterprise, supporting your key data security and compliance initiatives. the following are the solutions:

- Regulatory compliance
- Cloud storage
- Secure storage consolidation
- Multi-tenant solution providers
- Secure backup
- Intellectual property protection
- Secure information sharing

B. Transactions Logs

Transaction logs are a vital yet often overlooked component of database architecture. They are often forgotten because they are not something actively maintained like the schema contained within a database. In this article we'll examine how transaction logs are used in Microsoft SQL Server, maintenance and potential problems with them, how they can be used to restore a database, and finally, optimizing them for performance.

i. Big data transaction logs

A transaction log is a sequential record of all changes made to the database while the actual data is contained in a separate file. The transaction log contains enough information to undo all changes made to the data file as part of any individual transaction. The log records the start of a transaction, all the changes considered to be a part of it, and then the final commit or rollback of the transaction. Each database has at least one physical transaction log and one data file that is exclusive to the database for which it was created.

Figure 1. The growing volume, variety, and velocity of data

requires new methods of managing threats

Big data keeps a buffer of all of the changes to data for performance reasons. It writes items to the transaction log immediately, but does not write changes to the data file immediately. A checkpoint is written to the transaction log to indicate that a particular transaction has been completely written from the buffer to the data file. When Big data is restarted, it looks for the most recent checkpoint in the transaction log and rolls forward all transactions that have occurred from that point forward since it is not guaranteed to have been written to the data file until a checkpoint is entered in the transaction log. This prevents transactions from being lost that were in the buffer but not yet written to the data file.

ii. Transaction log maintenance

Transaction logs can present problems because they are often forgotten about until an issue occurs. The log continues to grow as operations are performed within the database. While the log continues to grow, the available disk space decreases. Unless routine action is taken to prevent it, the transaction log will eventually consume all available space allocated to it. If the log is configured to grow indefinitely as is the default, it will grow to consume all available physical disk space where it is stored. Either scenario causes the database to stop functioning.

Regular backups of the transaction log will help prevent it from consuming all of the disk space. The backup process truncates old log records no longer needed for recovery. The truncation process involves marking old records as inactive so they can be overwritten, which prevents the transaction log from growing too large. If frequent backups are not made, then the database should be configured with the "simple recovery model" option. The simple recovery model will force the transaction log to be truncated automatically each time a checkpoint is processed.

The truncation process that occurs as a result of a backup or the checkpoint will mark old log records as inactive so they can be overwritten, but it does not reduce the actual disk space allocated to the transaction log. The logs will keep the space allocated even if it is not used. This is where shrinking comes into the maintenance picture. A log is shrunk when a DBCC SHRINKDATABASE statement is executed against the owning database, a DBCC SHRINKFILE is executed against the specific transaction log, or an auto shrink operation occurs if it is enabled on the database. When shrinking the log, it is first truncated to mark inactive records and then the inactive records are removed. Depending upon how you are trying to shrink the log, you may or may not see immediate results. Ideally, shrinking should be performed on a scheduled basis so that it is forced to occur at points when the database utilization islower.

iii. Restoring a database

Transaction log backups can be used to restore a database to a specific point in time. A transaction log backup alone is not sufficient to restore a database. A backup of the data file is required as well. The data file backups are applied first. A full data file backup is restored and followed by any differential

backups of the data file. It is very important when restoring the data file backup not to mark the recovery as complete, otherwise no transaction log backups can be restored. Once the data file restore is complete, the backups of the transaction logs are applied to return the database to its desired state. If there have been multiple transaction log backups since the last database backup, then the transaction log backups need to be restored in the order in which they were created. There is another process known as *log shipping* that can be used to keep a hot stand-by of a database available. When log shipping is configured, a full backup of the database is copied to another server. From that point forward, transaction logs are periodically sent to the other server where they are automatically restored in the stand-by database. This keeps a hot spare of the server available. It is also commonly used to keep a reporting server up to date with recent data changes. Another server, known as a *monitor server*, can be configured to watch the shipping to make sure that it occurs on the scheduled interval. If it does not occur for some reason, then the monitor server will record an event to the event log. This makes log shipping a popular choice to be included as a part of disaster recovery planning.

iv. Optimizing for performance

Transaction logs play a vital role in the function of a database. As a result, they can have a direct impact on the overall system performance. There are certain configurations that can be made that will optimize the performance of transaction logs. The transaction log is a sequential write to the physical disk, and there are no reads that occur as a part of the logging process. Thus, if the logs are isolated on a separate disk, it will optimize the performance because there will be nothing interfering with the writing of the transaction log. Another optimization relates to the growth of the transaction log size. The log can be configured to grow as a percentage of the total size or at a set physical rate. Regardless of the growth option, the size of the growth should be large enough to prevent the log from needing to continually expand. If the growth rate is set to *low* the log may be forced to continually expand,

IV BIG DATA PRESENTS A NEW SECURITY CHALLENGE

Big data originates from multiple sources including sensors used to gather climate information, posts to social media sites, digital pictures and videos, purchase transaction records, and cell phone GPS signals, to name a few. Thanks to cloud computing and the socialization of the Internet, petabytes of unstructured data are created daily online and much of this information has an intrinsic business value if it can be captured and analyzed.

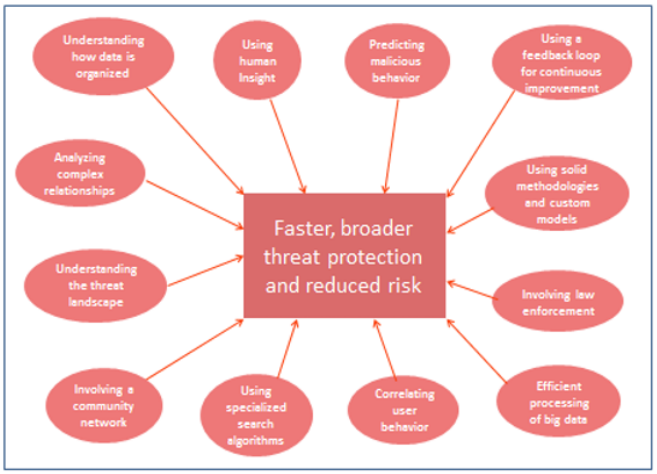

Figure 2. Core components of successful threat protection in the age of big data

For example, mobile communications companies collect data from cell towers; oil and gas companies collect data from refinery sensors and seismic exploration; electric power utilities collect data from power plants and distribution systems. Businesses collect large amounts of user-generated data from prospects and customers including credit card numbers, social security numbers, data on buying habits and patterns of usage.

The influx of big data and the need to move this information throughout an organization has created a massive new target for hackers and other cybercriminals. This data, which was previously unusable by organizations is now highly valuable, is subject to privacy laws and compliance regulations, and must be protected.

V. MANAGEMENT ISSUES

Management will, perhaps, be the most difficult problem to address with big data. This problem first surfaced a decade ago in the UK eScience initiatives where data was distributed geographically and “owned” and “managed” by multiple entities. Resolving issues of access, metadata, utilization, updating, governance, and reference (in publications) have proven to be major stumbling blocks. Unlike the collection of data by manual methods, where rigorous protocols are often followed in order to ensure accuracy and validity, digital data collection is much more relaxed. The richness of digital data representation prohibits a bespoke methodology for data collection. Data qualification often focuses more on missing data or outliers than trying to validate every item. Data is often very fine-grained such as click stream or metering data. Given the volume, it is impractical to validate every data item: new approaches to data qualification and validation are needed.

The sources of this data are varied - both temporally and spatially, by format, and by method of collection. Individuals contribute digital data in mediums comfortable to them: documents, drawings, pictures, sound and video recordings, models, software behaviors, user interface designs, etc – with or without adequate metadata describing what, when, where, who, why and how it was collected and its provenance. Yet, all this data is readily available for inspection and analysis. Going forward, data and information provenance will become a critical issue. JASON has noted [10] that "there is no universally accepted way to store raw data, … reduced data, and … the code and parameter choices that produced the data." Further, they note: "We are unaware of any robust, open source, platform independent solution to this problem." As far as we know, this remains true today. To summarize, there is no perfect big data management solution yet. This represents an important gap in the research literature on big data that needs to be filled.

VI. PROCESSING BIG DATA

Analytics Challenges Processing big data is a major challenge, perhaps more so than the storage or management problem. There are many types of analytics: descriptive, estimative, predictive, and prescriptive, leading to various types of decision and optimization models. Some common business analytics are depicted in Figure 1. Kaisler [11] presents another decomposition of analytics into 16 categories based on the types of problems to be addressed, including econometric models, game theory, control theory, evolutionary computation, and simulation models. The new normal is agile, advanced, predictive analytics that adapt readily to changing data sets and streams and yield information and knowledge to improve services and operations across academia, industry, and government. 3.1 Scaling A critical issue is whether or not an analytic process scales as the data set increases by orders of magnitude. Every algorithm has a "knee" – the point at which the algorithm's performance ceases to increase linearly with increasing computational resources and starts to plateau or, worse yet, peak, turn over, and start decreasing. Solving this problem requires a new algorithm for the problem, or rewriting the current algorithm to "translate" the knee farther up the scale. An open research question is whether for any given algorithm, there is a fundamental limit to its scalability. These limits are known for specific algorithms with specific implementations on specific machines at specific scales. General computational solutions, particularly using unstructured data, are not yet known. Table 8 gives some examples of analytic approaches that may not scale linearly. Simplistically, the processing of big data

VII. CONCLUSION

In this paper, have highlighted the Secure Data Storage and Transactions logs, by providing the auto-tiering mechanism in big data while placing the data in the data storage / transferring the data will be secure . hence unauthorized access can 't be tamper.

REFERENCES

[1] Ren, Yulong, and Wen Tang. "A Service Integrity Assurance Framework for Cloud Computing Based on Mapreduce."Proceedings of IEEE CCIS2012. Hangzhou: 2012, pp 240 – 244, Oct. 30 2012-Nov. 1 2012.

[2] N, Gonzalez, Miers C, Redigolo F, Carvalho T, Simplicio M, de Sousa G.T, and Pourzandi M. "A Quantitative Analysis of Current Security Concerns and Solutions for Cloud Computing.". Athens: 2011., pp 231 – 238, Nov. 29 2011-Dec. 1 2011

[3] Hao, Chen, and Ying Qiao. "Research of Cloud Computing based on the Hadoop platform.". Chengdu, China: 2011, pp. 181 – 184, 21-23 Oct 2011.

[4] Y, Amanatullah, Ipung H.P., Juliandri A, and Lim C. "Toward cloud computing reference architecture: Cloud service management perspective.". Jakarta: 2013, pp. 1-4, 13-14 Jun. 2013.

[5] A, Katal, Wazid M, and Goudar R.H. "Big data: Issues, challenges, tools and Good practices.". Noida: 2013, pp. 404 – 409, 8-10 Aug. 2013.

[6]. Foster I, Zhao Y, Raicu I, Lu S (2008) Cloud Computing and Grid Computing 360-Degree Compared. In: Grid Computing Environments Workshop (GCE'08). oi:10.1109/GCE.2008.4738445

[7]. Fellowes, W. (2008). Partly Cloudy, Blue-Sky Thinking about Cloud Computing. White paper. 451 Groups.

[8]. M. Casassa-Mont, S. Pearson and P. Bramhall, "Towards Accountable Management of Identity and Privacy: Sticky Policies and Enforceable Tracing Services", Proc. DEXA 2003, IEEE Computer Society, 2003, pp. 377-382

[9]. https://www.pcisecuritystandards.org/index.shtml

[10].http://en.wikipedia.org/wiki/Payment_Card_Industry_Data_Security_Standard, 24 January 2010

[11]. J. Salmon, "Clouded in uncertainty – the legal pitfalls of cloud computing", Computing, 24 Sept 2008, http://www.computing.co.uk/computing/features/2226701/clouded-uncertainty-4229153

[12]. Khajeh-Hosseini, A., Greenwood, D., Sommerville, I., (2010). Cloud Migration: A Case Study of Migrating an Enterprise IT System to IaaS. Submitted to IEEE CLOUD 2010

[13]. S. Overby, How to Negotiate a Better Cloud Computing Contract, CIO, April 21, 2010, http://www.cio.com/article/591629/How_to_Negotiate_a_Better_Cloud_Computing_Contract

[14]. Krautheim FJ (2009) Private virtual infrastructure for cloud computing. In: Proc of HotCloud

[15]. Santos N, Gummadi K, Rodrigues R (2009) Towards trusted cloud computing. In: Proc of HotCloud.

[16]. Arnon Rosenthal and Edward Sciore, View Security as the Basis for Data Warehouse Security

[17]. Arnon Rosenthal and Edward Sciore, View Security as the Basis for Data Warehouse Security, Proceedings of the International Workshop on Design and Management of Data Warehouse (DMDW'2000), Sweden, June, 2000.

[18]. Alan R. Downing, Ira B. Greenberg, and Teresa F. Lunt, ISSUES IN DISTRIBUTED DATABASE SECURITY

Mr. Yannam Apparao, Currently working as Associate Professor, Marri Laxman Reddy Institute of Technology and Management, Dundigal, Quthbullar(M),R.R.Distic, Telangana-500043.

Mrs. K.Laxminarayanamma, Currently working as Associate Professor, Institute of Aeronautical Engineering, Quthbullapur(M), R.R.Distic, Telangana, India. Affiliated to Jawaharlal Nehru Technological University,